Software Models for Mobile Wireless Computing

Evaggelia Pitoura

Computer Science Department,
University of Ioannina,
Ioannina, Greece
http://www.cs.uoi.gr/~ pitoura

Summer School, Jyvaskyla, August 1998

Introduction

Restrictions [25, 7, 10, 2]:

Mobility

- System configuration is no longer static: the center of activity, the topology, the system load, and locality, change dynamically
- need to search to locate objects
- various forms of heterogeneity

Wireless Communications

- offer less bandwidth
- more expensive
- less reliable
- Consequently, connectivity is weak and often intermittent

Portable Devices

- light and small to be easily carried around
- Such considerations, in conjunction with a given cost and level of technology ⇒ mobile elements with less resources (e.g., memory, screen size and disk capacity)
- reliance on battery
- can be more easily accidentally damaged, stolen, or lost, thus, less secure and reliable

Introduction

What type of functionality should be assigned to mobile hosts?

Mobile units are still characterized as:

- unreliable and prone to hard failures, i.e., theft, loss or accidental damage,
- resource-poor relative to static hosts.

These reasons justify treating the mobile units as dumb terminals running just a user-interface and off-load all functionality from the mobile unit to the fixed network

Examples: InfoPad [16] and ParcTab [28] projects

On the other hand, slow and unreliable networks argue for putting additional functionality at the mobile hosts to lessen their dependency to remote servers.

Although, there is no consensus yet on the specific role mobile hosts will play in distributed computation, the above contradictory considerations lead to models that provide for a flexible adjustment of the functionality assigned to mobile hosts.

Adaptability

A mobile system is presented with resources of varying number and quality:

- Connectivity conditions vary from total disconnections to full connectivity
- Available resources are not static either, for instance a "docked" mobile computer may have access to a larger display or memory.
- the location of mobile elements changes and so does the network configuration and the center of computational activity

Need to **adapt** to the constantly changing environmental conditions [13, 26, 12, 7].

How can adaptivity be captured and realized?

By varying the partition of duties between the mobile and static elements

Example: during disconnection, a mobile host may work autonomously, while during periods of strong connectivity, depend heavily on the fixed network sparing its scarce local resources.

By varying the quality of data available at the mobile host

Example: Fidelity [18]: the degree to which a copy of data presented for use at a site matches the reference copy at the server. Fidelity has many dimensions

Adaptability

Where should support for mobility and adaptivity be placed.

Should applications be aware of their environment?

Strategies range between two extremes [25, 18]:

At one extreme, adaptivity is solely the responsibility of the underlying system and is performed transparently from applications.

+ existing applications continue to work unchanged.
- since there is no single best way to serve applications with diverse needs, this approach may be inadequate or even make performance worse than providing no support for adaptivity at all.
- may no be attainable, for instance, during periods of long disconnections

At the other extreme, adaptation is left entirely to individual applications. No support is provided by the operating system.

- No focal point to resolve the potentially incompatible resource demands of different applications or to enforce limits on resource usage of resources
- applications must be written anew which can become very complicated.

Application-aware [27] support for mobility lies in between

Adaptability

Application-Awareness

Additional requirements [1]

1. A mechanism to **monitor** the level and quality of resources and **inform applications** about any relevant changes in their environment

Environmental changes include: changes of the location and of resource availability (such as bandwidth, memory, etc)

Raises a lot of issues.

2. Applications must be **agile** [19, 1], that is able to receive events in an asynchronous manner and react appropriately.

3. There is a need for a **central point** for managing resources and authorizing any application-initiated request for their use

In [33], changes in the environment are modeled as asynchronous events which are delivered to the application. Detection either within the kernel or at the user-level. Detection of an event is decoupled from its delivery.

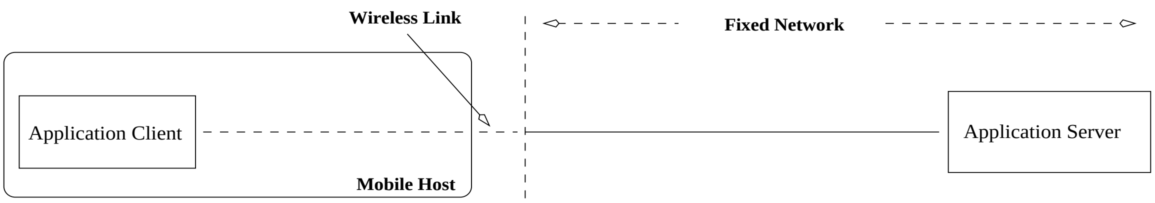
In Odyssey [17, 27, 18], the application negotiates and registers a window of tolerance with the system for a particular resource. If availability of that resource rises above or falls below the limits in the window, Odyssey notifies the application. Once notified, it is the application's responsibility to adapt its behavior accordingly.

Handling mobility spans multiple levels

Mobile Client/Server Models

An application component executing in one computing system, called the *client* requests a service from an application component executing in another computing system, called the *server*

The mobile host acts as the client requesting services from servers located at the fixed network



Multiple servers or replicated servers

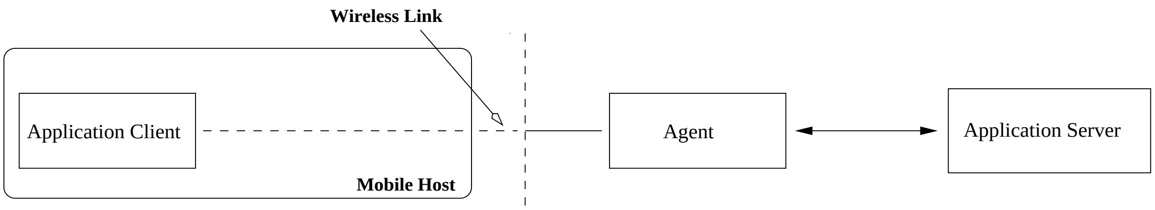
Multiple Server Architectures
Service Handoff [11] When multiple interconnected servers: attach the client server located closest to it.

In [29, 14], the mapping of clients to servers is completely transparent to the application and is taken care of by an underlying coherence control scheme among the servers. In contrast, in Bayou [31] application-level

Assumptions: static clients, reliable and fast communications, and relatively resource-rich and reliable clients

Client/Agent/Server Architectures

A three-tier or client/agent/server model [4, 8, 20, 35, 30] that uses messaging and queuing infrastructure for communications from the mobile client to the agent and vice-versa



Agents are used in a variety of forms and roles

Agents as Proxies

The agent acts as a complete surrogate or *proxy* of the mobile host on the fixed network.

Any communication to and from the mobile host goes through its agent.

Can be generalized by having an agent acting as a surrogate of multiple mobile hosts [8].

Service-Specific Agents

The agent provides mobile-aware access to specific services or applications, e.g., web browsing [9] or database access [20].

Any client's request and server's reply associated with the specific service through the agent.

Per mobile host as many agents as the services it needs access to.

Client/Agent/Server Architectures

The agent performs **various functions:**

- messaging and queuing
- various optimizations for weak connectivity
  manipulate the data prior to their transmissions to the client [35, 8, 30], by changing their transmission order so that the most important information is transfered first, by performing data specific lossy compression tailored to the client, or by batching together multiple replies.
- a more active role [20, 30], e.g., notify the client when predefined events occur
- to offload functionality form the client, start/stop specific functions at the mobile unit or execute client specific services.
- Disconnected operation and agents

Position of the agent: depends on its role

At the fringe of the fixed network, i.e., at the base station (especially when surrogate) [35, 4]: easier to gather information about the wireless link; special link level protocol between the mobile host and the agent; personalized information available.

But, the agent may need to move along with its mobile host, or the current base station may not be trustworthy.

When service-specific agents, either closer to the majority of their clients or closer to the server.

Client/Agent/Server Architectures

Relocation of the agents at the fixed network

When?

Need to manage the location of the agent. A mobile motion prediction algorithm to predict the future location of a mobile user and place a new proxy [15]

Summary

+ alleviates the impact of the limited bandwidth and the poor reliability of the wireless link by continuously maintaining the client's presence on the fixed network via the agent.

+ splits the interaction between mobile clients and fixed servers in two parts: one between the client and the agent, and one between the agent and the server. Different protocols for each part of the interaction and each part of the interaction may be executed independently of the other.

- fails to sustain the current computation at the mobile client during periods of disconnection.

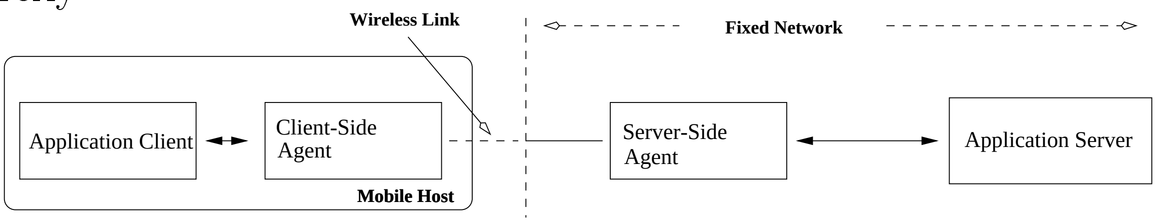
- requires changes to the client code for the development of the client/agent interaction

- the agent can directly optimize only data transmission over the wireless link from the fixed network to the mobile client and not vice versa.

The Pair of Agents Model

A client-side agent to reduce data transmission, improve availability and sustain any computation at the mobile host uninterrupted [24, 9]

from the point of view of the client: a coresident local server proxy



- variety of optimizations:
  background prefetching
  relocate computation between the agents
- clear separation of responsibilities
- client interoperability
- flexibility in handling disconnections. e.g., a local cache at the client
- communication protocol that can facilitate highly effective data reduction and protocol optimization

relatively heavy-weight clients

development work both at the server and at the client site. It suffices a different pair of agents per application type

The Pair of Agents Model

Proxy pairs have been gaining attention [35, 8].

Extensions to RPC [12, 5, 3],

e.g., asynchronous queued RPC [12],

– The RPC is stored in a local stable log at a client-side agent and control is immediately returned to the application.

– When connected, the log is drained in the background – Queueing RPCs leaves space for performing various optimizations on the log.

– Delivering replies from the server may require multiple retries [12, 5].

IBM's WebExpress [9] for optimizing web browsing in a wireless environment.

Peer-to-Peer Models

Applications in which the server resides at a mobile host

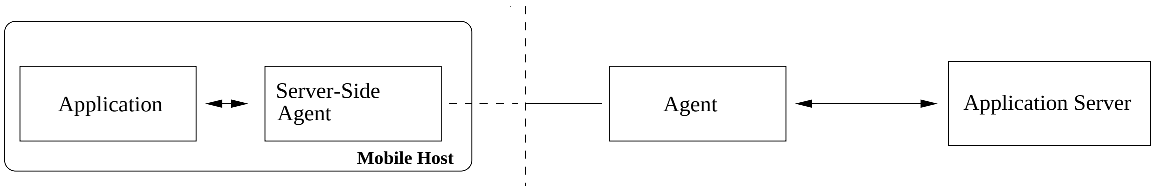
Example: cooperative work on some data using their portable computers [23]

Ideally, each site has the full functionality of both a client and a server. In this case, mobile hosts are equal partners in distributed computations.

Heavy-weight mobile hosts.

Disconnections in addition make the server unavailable to clients.

To deal with disconnections and weak connectivity a server-side intercept agent on the mobile host as well.



Server-side agent at the mobile host posses special features: e.g., a mechanism to automatically start applications on demand [3].

Mobile Agents

Besides the functional components, organization of data.

As a collection of objects: the units of information exchange among mobile and static hosts.

Objects encapsulate not only pure data but also information for their manipulation, e.g., operations.

Can be built on top of an existing database or file system.

In the *Rover* toolkit [12]: relocatable dynamic objects (RDO).

In the *Pro-motion* infrastructure [32]: *compact* units of caching and replication.

Active computations + Mobility $\Rightarrow$ *mobile agents models*.

Mobile agents: processes along with data dispatched from a source computer to accomplish a specified task [6, 34].

- After submission, proceeds autonomously and independently of the sending client.
- When it reaches a server, delivered to an agent execution environment. If necessary authentication credentials, it is executed parts are started.
- Transport itself to another server, spawn new agents, or interact with other agents.
- Upon completion, delivers the results to the sending client or to another server.

Mobile Agents

Support intermittent connectivity, slow networks, and light-weight devices makes them appropriate for wireless mobile computing [6, 21].

Disconnected operation: during a brief connection interval, a mobile client submits an agent to the fixed network.

Conversely, a mobile agent loaded from the fixed network onto a laptop before disconnection. The agent acts as a surrogate for the application allowing interaction even during disconnections.

Weak connectivity: the overall communication traffic through the wireless link reduced from a possibly large number of messages to the submission of a single agent and its result.

Shift the burden of computation from resource-poor mobile hosts to the fixed network.

Mobility is inherent in the model. Mobile agents migrate to find resources but or follow their clients.

Adaptability

Relation to the client/server model and its extensions.

1. Implement the agent as mobile or

2. A client submits a mobile agent to the (static) agent that refines it before launching it to servers on the network. Thus, agents with capabilities to process mobile agents ⇒ *programmable* agents. Current research on *active networks* [30].

Summary

Agents placed between the mobile client and the fixed server

- to alleviate the constraints of the wireless link, by performing various communication optimizations,
- to handle resource constraints, by undertaking part of the functionality of resource-poor mobile clients.

At what level do agents function?

Multiple agents at different levels that cooperate. ways.

Agents at lower layers convey information to agents at higher layers and vice versa.

Another approach: agents, called *filters*, that operate on protocols [35] rather than at the application or operating system level.

Example: an MPEG-agent or a TCP-agent.

Fewer protocols than applications thus less development work. Applications control agents by turning them on and off.

Surrogate of the Mobile Host	All functionality of the host is off-loaded to the agent
Filter	Implements the mobile-specific part of a protocol: E.g., an MPEG-1 filter that discards MPEG frames, or an RPC filter that implements asynchronous queueing RPC
Service-specific	Implements the mobile-specific part of a service/application E.g., an Exmh-agent, a web-agent, or a file-system agent
Programmable	Understands and can process code or mobile agents sent by clients or servers

References

[1] A. Acharya, M. Ranganathan, and J. Saltz. Sumatra: A Language for Resource-Aware Mobile Programs. In J. Vitek and C. Tschudin, editors, *Mobile Object Systems*, pages 571–594. Springer Verlag LNCS, 1998. To appear.

[2] R. Alonso and H. F. Korth. Database System Issues in Nomadic Computing. In *Proceedings of the 1993 SIGMOD Conference*, Washington, D.C., May 1993.

[3] A. Athan and D. Duchamp. Agent-Mediated Message Passing for Constrained Environments. In *Proceedings USENIX Symposium on Mobile and Location-Independent Computing*, pages 103–1070, Cambridge, Massachusetts, August 1993.

[4] B. R. Badrinath, A. Bakre, T. Imielinski, and R. Marantz. Handling Mobile Clients: A Case for Indirect Interaction. In *Proceedings of the 4th Workshop on Workstation Operating Systems*, Aigen, Austria, October 1993.

[5] A. Bakre and B.R. Badrinath. Implementation and Performance Evaluation of Indirect TCP. *IEEE Transactions on Computers*, 46(3), March 1997.

[6] D. Chess, B. Grosof, C. Harrison, D. Levine, C. Parris, and G. Tsudik. Itinerant Agents for Mobile Computing. *IEEE Personal Communications*, 2(5), October 1995.

[7] G. H. Forman and J. Zahorjan. The Challenges of Mobile Computing. *IEEE Computer*, 27(6):38–47, April 1994.

[8] A. Fox, S. D. Gribble, E. A. Brewer, and E. Amir. Adapting to Network and Client Variability via On-Demand Dynamic Distillation. In *Proceedings of the ASPLOS-VII*, Cambridge, MA, October 1996.

[9] B C. Housel, G. Samaras, and D. B. Lindquist. WebExpress: A Client/Intercept Based System for Optimizing Web Browsing in a Wireless Environment. *ACM/Baltzer Mobile Networking and Applications (MONET)*, 1997. Special Issue on Mobile Networking on the Internet. To appear. Also, University of Cyprus, CS-TR 96-18, December 1996.

[10] T. Imielinksi and B. R. Badrinath. Wireless Mobile Computing: Challenges in Data Management. *Communications of the ACM*, 37(10), October 1994.

[11] R. Jain and N. Krishnakumar. Network Support for Personal Information Services for PCS Users. In *Proceedings of the IEEE Conference on Networks for Personal Communications*, March 1994.

[12] A. D. Joseph, J. A. Tauber, and M. F. Kaashoek. Mobile Computing with the Rover Toolkit. *IEEE Transactions on Computers*, February 1997.

[13] R. H. Katz. Adaptation and Mobility in Wireless Information Systems. *IEEE Personal Communications*, 1:6–17, 1994.

[14] J. J. Kistler and M. Satyanarayanan. Disconnected Operation in the Coda File System. *ACM Transactions on Computer Systems*, 10(1):213–225, February 1992.

[15] G. Y. Liu, A. Marlevi, and G. Q. Maguire Jr. A Mobile Virtual-Distributed System Architecture for Supporting Wireless Mobile Computing and Communications. *Wireless Networks*, 2:77–86, 1996.

[16] S. Narayanaswamy, S. Seshan, and et. al. Application and Network Support for InfoPad. *IEEE Personal Communications Magazine*, March 1996.

[17] B. Noble and M. Satyanarayanan. A Research Status Report on Adaptation for Mobile Data Access. *Sigmod Record*, 24(4), December 1995.

[18] B. D. Noble, M. Price, and M. Satyanarayanan. A Programming Interface for Application-Aware Adaptation in Mobile Computing. *Computing Systems*, 8(4), Winter 1996.

[19] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R. Walker. Agile Application-Aware Adaptation for Mobility. In *Proceedings of the 16th ACM Symposium on Operating System Principles*, October 1997.

[20] Oracle. Oracle Mobile Agents Technical Product Summary. www.oracle.com/products/networking/mobile_agents/html/, June 1997.

[21] E. Pitoura and B. Bhargava. A Framework for Providing Consistent and Recoverable Agent-Based Access to Heterogeneous Mobile Databases . *ACM SIGMOD Record*, 24(3):44–49, September 1995.

[22] E. Pitoura and G. Samaras. *Data Management for Mobile Computing*. Kluwer Academic Publishers, 1998.

[23] P. Reiher, J. Popek, M. Gunter, J. Salomone, and D. Ratner. Peer-to-Peer Reconiliation Based Replication for Mobile Computers. In *Proceedings of the European Conference on Object Oriented Programming 2nd Workshop on Mobility and Replication*, June 1996.

[24] G. Samaras and A. Pitsillides. Client/Intercept: a Computational Model for Wireless Environments. In *Proceedings of the 4th International Conference on Telecommunications (ICT'97)*, Melbourne, Australia, April 1997.

[25] M. Satyanarayanan. Fundamental Challenges in Mobile Computing. In *Proceedings of the 15th ACM Symposium on Principles of Distributed Computing*, Philadelphia, PA, May 1996.

[26] M. Satyanarayanan. Mobile Information Access. *IEEE Personal Communications*, 3(1), February 1996.

[27] M. Satyanarayanan, B. Noble, P. Kumar, and M. Price. Application-Aware Adaptation for Mobile Computing. *Operating System Review*, 29(1), January 1995.

[28] B. N. Schilit, N. Adams, R. Gold, M. Tso, and R. Want. The ParcTab Mobile Computing System. In *Proceedings of the 4th IEEE Workshop on Workstation Operating Ssytems (WWOS-IV)*, pages 34–39, October 1993.

[29] C. D. Tait and D. Duchamp. Service Interface and Replica Management Algorithm for Mobile File System Clients. In *Proceedings of the First International Conference on Parallel and Distributed Information Systems*, pages 190–197, 1991.

[30] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, and G. J. Minden. A Survey of Active Network Research. *IEEE Communication Magazine*, 35(1):80–86, January 1996.

[31] D. B. Terry, M. M Theimer, K. Petersen, A. J. Demers, M. J Spreitzer, and C. H. Hauser. Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System. In *Proceedings of the 15th ACM Symposium on Operating Systems Principles*, December 1995.

[32] G. Walborn and P. K. Chrysanthis. PRO-MOTION: Support for Mobile Database Access. *Personal Technologies Journal*, 1(3), September 1997.

[33] G. Welling and B. R. Badrinath. A Framework for Environment Aware Mobile Applications. In *Proceedings of the IEEE International Conference on Distributed Computing Systems*, 1997.

[34] J. E. White. Mobile Agents. General Magic White Paper, www.genmagic.com/agents, 1996.

[35] B. Zenel and D. Duchamp. General Purpose Proxies: Solved and Unsolved Problems. In *Proceedings of the Hot-OS VI*, 1997.